

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Online 3D multiplayerová hra

Online 3D Multiplayer Game

Zadání bakalářské práce

Student: **Vladimír Bednář**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Online 3D multiplayerová hra**
Online 3D Multiplayer Game

Jazyk vypracování: čeština

Zásady pro vypracování:

Člověk je již od nepaměti tvor hravý a v dnešní době tomu není jinak. Proto také neustále vznikají nové počítačové hry. V oblibě jsou především 3D online hry, kdy lidé bojují či soupeří mezi sebou nebo kooperují proti něčemu či někomu.

Cílem práce bude vytvořit multiplayerovou hru za použití například Unity enginu. Součástí bude také uživatelská a programátorská dokumentace.

Body zadání:

1. Seznámení se s problematikou vývoje multiplayerové hry.
2. Seznámení a popis běžně používaných herních enginů.
3. Návrh hry, vlastní 3D grafika a textury.
4. Návrh spojení server/klient, klient/server, UML diagramy.
5. Implementace serveru a klienta hry.
6. Zhodnocení výkonu aplikace na základě testů s hráči

Seznam doporučené odborné literatury:


- [1] Technologies, U. (2015). Unity - Manual: Unity Manual. [online] Docs.unity3d.com. Available at: <https://docs.unity3d.com/Manual/index.html>
- [2] BARR, Pippin, James NOBLE a Robert BIDDLE. Video game values: Human–computer interaction and games. Interacting with Computers [online]. 2007, 19(2), 180-195. DOI: 10.1016/j.intcom.2006.08.008. ISSN 09535438. Dostupné z: <https://academic.oup.com/iwc/article-lookup/doi/10.1016/j.intcom.2006.08.008>
- [3] Manker, Jon & Arvola, Mattias. (2011). Prototyping in Game Design: Externalization and Internalization of Game Ideas. Proceedings of HCI 2011 - 25th BCS Conference on Human Computer Interaction.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radek Simkanič, DiS**

Datum zadání: 01.09.2019

Datum odevzdání: 30.04.2020


doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry




prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2020



.....

Rád bych poděkoval svému vedoucímu práce Ing. Radkovi Simkaničovi za rady, pomoc a vstřícnost při konzultacích a odborné vedení bakalářské práce. Také bych rád poděkoval Mgr. Ingrid Petrikové a Bc. Tereze Petrikové za pomoc s jazykovou stránkou této bakalářské práce.

Abstrakt

Tato práce se zabývá procesem vývoje počítačové hry a je v ní řešena i problematika vývoje hry pro více hráčů (multiplayer). Hra je implementována v jazyce C# v prostředí herního enginu Unity. Multiplayer je řešen architekturou klient-server. Ke hře je vytvořena i část grafického obsahu ve formě 3D modelů a textur. Součástí vývoje hry je testování s hráči, na základě jejichž výsledků je hra dále upravována. Postupným testováním a upravováním hry bylo dosaženo vytvoření funkční multiplayerové hry pro tři hráče.

Klíčová slova: Unity engine; multiplayer; C#; videohra

Abstract

This bachelor thesis deals with the process of development of PC multiplayer game. Game is implemented in C# programming language in Unity engine. Multiplayer is solved by client-server architecture. For this a game special graphical package of 3D models and textures was made. In a process of development this game was tested with players. Based on the results of the testing the game was modified. In the end a fully functional multiplayer game was developed.

Keywords: Unity engine; multiplayer; C#; video game

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
Seznam výpisů zdrojového kódu	12
1 Úvod	13
2 Seznámení s problematikou vývoje hry	14
2.1 Indie hry	15
2.2 Problematika hry více hráčů	16
3 Herní engine	17
3.1 Unity engine	17
3.2 Unreal engine	18
3.3 CryEngine	18
3.4 Srovnání herních engineů	19
4 Návrh multiplayerové hry	20
4.1 Volba softwaru	20
4.2 Návrh síťového propojení	20
5 Vlastní 3D grafika a textury	21
6 Implementace multiplayerové hry v Unity engineu	23
6.1 Konfigurace serveru a klienta hry	23
6.2 Připojení do hry	24
6.3 Aktualizace pozice vozidel	24
6.4 Detekce zásahu	25
6.5 Vykreslování laserového paprsku	27
6.6 Uživatelské rozhraní hry	28
6.7 Ostatní herní mechaniky	29
6.8 Design herní mapy	30
7 Testování hry	32
7.1 Metodika	32
7.2 První vlna testování	32
7.3 Úprava hry na základě první vlny testování	34

7.4	Druhá vlna testování	35
7.5	Úprava hry na základě druhé vlny testování	35
7.6	Třetí vlna testování	37
8	Závěr	39
	Literatura	40
	Přílohy	41
A	Přílohy	42
B	Hardwarové požadavky	43
B.1	Optimální hardwarové požadavky	43
B.2	Minimální hardwarové požadavky	43
C	Seznam použitého softwaru	44
D	Návod ke zprovoznění virtuální LAN	45
E	Výsledky rozhovorů s testery po testování hry	46

Seznam použitých zkratk a symbolů

AAA	– triple-A
3D	– trojrozměrný prostor
2D	– dvojrozměrný prostor
ID	– identifikační číslo
UI	– uživatelské rozhraní

Seznam obrázků

1	Unreal engine Blueprints	18
2	První návrh modelu kombajnu	21
3	Druhý návrh modelu kombajnu	21
4	UV mapování v programu Blender	22
5	Třídní diagram serializovatelných tříd pro přenos herních dat	24
6	První způsob detekce zásahu (snímek z kamery na děle)	26
7	Druhý způsob detekce zásahu (snímek z kamery na děle)	26
8	Zastavení laserového paprsku o překážku	28
9	Hlavní nabídka	29
10	Vyskakovací okno ve hře	29
11	Běžný režim střelby	29
12	Ostřelovací režim střelby	29
13	Mapa oblastí	30
14	První hra	33
15	Druhá hra	33
16	Třetí hra	33
17	Čtvrtá hra	33
18	Pátá hra	33
19	Šestá hra	35
20	Sedmá hra	35
21	Osmá hra	35
22	Devátá hra	36
23	Desátá hra	36
24	Rozptyl při střelbě	36
25	Jedenáctá hra	37
26	Dvanáctá hra	37
27	Třináctá hra	37
28	Čtrnáctá hra	38
29	Patnáctá hra	38

Seznam tabulek

1	Srovnání herních enginů [8, 6, 7, 9]	19
2	Výsledky rozhovorů v první vlně testování	47
3	Výsledky rozhovorů ve druhé vlně testování	48
4	Výsledky rozhovorů ve třetí vlně testování	49

Seznam výpisů zdrojového kódu

1	Ukázka konfigurace klienta hry a serveru	23
2	Řešení problému se statickými proměnnými	34

1 Úvod

Hraní her se v dnešní době stalo běžnou součástí života mnoha lidí podobně jako sledování filmů, čtení knih nebo novin. Díky stále rychlejšímu internetovému připojení je již dlouhou dobu trendem v oblasti hráčství **multiplayer**, neboli hra pro více hráčů. Cílem této práce je vytvoření multiplayerové hry. Nabídka multiplayerových her je velmi pestrá, tudíž je třeba, aby hra potencionální hráče něčím zaujala. Tématem hry proto budou kombajny vybavené zbraněmi, které budou bojovat o úrodu na poli. Zvítězí ten, kdo nejrychleji sklídí dané množství obilí. Celý herní zážitek by měly zpestřit zmíněné zbraně, jejichž použití rozhodně nebude proti pravidlům. Ideálním vyústěním je bitva hráčů, která je zabaví na několik hodin.

V počáteční fázi práce bude provedena komparace vývoje her na profesionální úrovni a vývoje her nezávislými programátory a grafiky pro vlastní potřebu. Bude provedena analýza procesu vývoje, a tak dojde k odhalení podobností či odlišností v jejich práci. Cílem komparace je inspirovat se pro účely vývoje vlastního projektu. Dalším logickým krokem je zjištění, jakým způsobem jsou tyto hry dále spravovány a monetizovány. Budou prozkoumána i specifika multiplayerových her.

Náplní práce bude seznámení se s vývojovým prostředím Unity enginu a jeho klíčovými vlastnostmi. Práce se zaměří na konkurenční herní enginy, jejich klíčové vlastnosti a využití ve videoherním průmyslu. Výstupem této sekce bude porovnání zmíněných enginů na základě kritérií, které budou zjištěny jako klíčové pro práci na tomto projektu.

Před zahájením práce na implementaci vlastní hry bude vytvořen návrh klíčových komponent výsledného produktu. Mezi tyto komponenty bude mimo jiné patřit komponenta pro přenos informací po síti a návrh uživatelsky přívětivého uživatelského rozhraní.

V projektu tohoto charakteru bude nutné vytvořit část grafického obsahu specifického pro výslednou hru. Konkrétně se jedná o kombajn vybavený zbraňovým arzenálem a jiné 3D modely, které nejsou volně dostupné na internetu. Jako grafické editory budou použity programy **Blender**, **Paint.net** a **Krita**. Programy byly vybrány na základě předchozích kladných zkušeností autora projektu.

Výstupem bude multiplayerová hra, která zohlední výsledky analýz a komparací jednotlivých fází. Součástí vývoje bude vyhodnocování dat získaných od respondentů pomocí strukturovaných rozhovorů. Další část dat se získá z informací ukládaných o hrách pomocí aplikace na straně klienta nebo serveru. Jedná se o data týkající se pohybu hráčů po herní mapě.

2 Seznámení s problematikou vývoje hry

Videohry jsou nejvýdělečnější součástí zábavního průmyslu. Za jejich vývojem stojí primárně velká a dobře financovaná vývojářská studia, která se snaží ke každé hře přistupovat inovativně a vkládat do ní nové herní mechaniky, a tím si zajistit konkurenční výhodu na trhu. Herní tituly, do kterých bylo investováno velké množství financí a byly vyvíjeny zkušeným vývojářským studiem za účelem velkého zisku, se nazývají **AAA herní tituly** (*triple-A game titles*). [1] Často však hry tvoří i nezávislí nadšenci. Jejich hry se označují jako nezávislé počítačové hry. V odvětví herního průmyslu jsou téměř výhradně označovány jako **indie hry** (odvozeno od anglického slova *independent*). [2]

Hry mohou být vyvíjeny pro více zařízení. Mezi tato zařízení patří například mobilní telefony, počítače a herní konzole. Tato zařízení mohou mít různé operační systémy (dále jen OS). Libovolná kombinace hardwaru a OS se označuje jako **platforma**. Vývoj hry pro více platforem zvyšuje komerční potenciál. Výsledná hra se označuje jako **multiplatformní hra**. Multiplatformní hra musí splňovat hardwarové požadavky nejméně výkonné platformy, pro kterou je vyvíjena, v důsledku čehož nemusí splnit očekávání zákazníků disponujících dražším a výkonnějším hardwarem v oblasti grafického zpracování. [1]

Vývojářská studia mohou být různě organizována na základě preferencí vedení firmy, a to například jako **superstudio**, kde týmy odborníků pracují na více projektech současně. Další možností jsou zaměstnanci (různých odborností) rozděleni do týmů, kde každý tým pracuje na jednom herním titulu. Obě tato rozvržení obsahují dělení jednotlivých týmů do základních kategorií – grafici, designéři, programátoři a produkce. Součástí těchto týmů jsou zaměstnanci, jejichž působnost se překrývá. Patří mezi ně animátoři, kteří se zabývají tvorbou animací. Animace se v dnešní době často spojuje se zaznamenáváním pohybu skutečných osob a následným převodem tohoto záznamu jako animace do hry. Tato technika se nazývá **motion tracking**. Další speciální roli v týmu mají takzvaní **riggeři**, kteří se zabývají mapováním kloubů na modely postav (**rigování**), aby bylo možné model v těchto kloubech ohýbat. Mezi další zaměstnance s překrývající se působností patří i techničtí umělci, jejichž náplní práce je jak programování, tak i tvorba grafického obsahu. [1]

Proces vývoje začíná nápadem, který může být inovativní, novou herní mechanikou, příběhem nebo jakýmkoliv jiným aspektem, který by mohl zaujmout potencionální hráče. Nezřídka bývá hra vyvíjena na základě populární knihy nebo filmu. Dalším krokem, pokud vedení společnosti uzná vývoj titulu za vhodný, je sestavení dokumentu popisujícího výsledný produkt – klíčové herní mechaniky, postavy, příběh a vizualizace v podobě náčrtů a modelů. Tento dokument se nazývá **concept book**. Tato publikace napomáhá vývojářům se detailně seznámit s projektem. [1]

Neméně důležitým krokem je volba životního cyklu vývoje softwaru. Velmi oblíbeným je vodopádový model, který se dá použít pro menší projekty s jasně definovaným výstupem. Jeho typickou vlastností je členění vývoje do několika kroků, ke kterým se nedá po jejich dokončení

vracet. Tato vlastnost může být vnímána jako nedostatek, ale i jako silná stránka (vzhledem k ušetřenému času při vývoji). Pro komplikovanější projekty může být použito rozšíření vodo-
pádového modelu, takzvaného agilního modelu. Tento model taktéž obsahuje sérii kroků, ale po každém kroku navíc následuje revize a úprava kroků předešlých. Pro mnohem komplexnější projekty se specifickými požadavky existují ještě modely, jako například **extrémní programování** (*extreme programming*), **RUP** (*rational unified process*), **mlýn** (*scrum*) a spousta dalších. [3]

Další otázkou, kterou je třeba zvážit před začátkem vývoje, je rozpočet. Kapitál může být ušetřen nejen na výdajích na zaměstnance, ale také volbou vhodného životního cyklu vývoje softwaru. Pro menší projekty je vhodné použít jednodušší životní cykly vývoje softwaru. Ušetřit značné množství času a peněz se dá i použitím již existujícího herního enginu. Mezi dnes dostupné, hotové herní enginy, patří v oblasti trojrozměrných her například **Unity engine**, **Unreal engine** nebo **CryEngine**. Pro menší, dvojrozměrné hry, často prohlížečové a mobilní, jsou oblíbenou volbou engine **Godot** nebo frameworky **Monogame** a **Phaser**. Na trhu je nepřeborné množství dalších herních enginů a frameworků, s jejichž pomocí je možné velmi rychle tvořit videohry. [3]

Práce na hře nekončí ani po uvedení hotového produktu na trh. Důležité je zvolit vhodnou metodu marketingové kampaně. Dobrých her je na trhu spousta, a tak je nutné dostatečně dobře propagovat vlastní produkt, aby se informace o něm dostaly k cílovým spotřebitelům. Velkou roli v prodejnosti hry hrají recenze kritiků, kterým se kopie hry na požádání často poskytují dříve než ostatním hráčům. V poslední době je trendem ke hrám po jejich vydání přidávat rozšíření, takzvaná **DLC** (*downloadable content*), která do hry přidávají nový obsah. DLC se prodávají zvlášť, nebo jsou dostupná zdarma za účelem udržet hráče. [3]

Dalším způsobem, jak vydělávat na hře, kromě jejího prodeje, jsou takzvané **mikrotransakce**. Tento pojem označuje platby ve hře za různá vizuální nebo funkční vylepšení. Na mikrotransakcích může být postaven celý komerční model hry. To se často děje u **free to play** her, které jsou k dostání zdarma, ale obsahují placená rozšíření, která vylepšují herní zážitek. Nejtypičtějším je tento model u her na mobilní telefony nebo u multiplayerových her na hrdiny (**MMORPG** – *massively multiplayer online role-playing game*). [3]

2.1 Indie hry

Jako indie hry se označují hry vyvíjené a distribuované nezávislými vývojáři nezávisle na velkých videoherních společnostech. [2, 3] Motivací k vývoji takových her nebývá zisk, ale často jde o kreativní vyjádření autorů. Na internetových fórech a sociálních sítích dnes existuje velká komunita nezávislých vývojářů. Členové se od sebe navzájem učí a nevnímají se jako konkurenci. Vývojáři v této oblasti herního průmyslu vytvářejí mezinárodní týmy, jejichž členové spolupracují na projektech, i když se nacházejí v různých časových pásmech. Komunikace je zajištěna prostřednictvím videokonferencí a chatů, a proto jsou soubory sdíleny online. [3]

Při online komunikaci může docházet k řadě problémů spojených s povědomím jednotlivých vývojářů o projektu. Menší komponenty softwaru nemusí být dostatečně promyšlené. Může chy-

být dostatek kritického pohledu na projekt, pokud jsou všichni členové týmu pohlceni svou prací a nevidí chyby ostatních vývojářů, které mohou mít dopad na kvalitu hry. [2]

Podobně jako u větších videoherních společností se i zde neobejde práce bez rozdělení základních rolí v týmu. Velkou roli hraje vedoucí projektu, který určuje směr vývoje hry. Dalo by se říct, že svou působností supluje concept book, který používají velká vývojářská studia pro jasnou definici cílového produktu. [2]

2.2 Problematika hry více hráčů

Nejefektivnějšími způsoby, jak implementovat multiplayer do své hry, jsou vlastní servery. Toto řešení však vyžaduje vlastnit dostatečně robustní hardware s přístupem k velmi rychlému internetovému připojení. Jako další možnost se nabízí, aby si skupinky hráčů zvolili mezi sebou jednoho hráče s dobrým hardwarem a internetovým připojením. Tento hráč pak může provozovat malý soukromý server pro ostatní, což je levné řešení vhodné pro vývojáře s nízkým rozpočtem.

Ve hrách pro jednoho hráče dochází k podvádění například pomocí "cheat kódů"¹ umístěných do hry samotnými vývojáři nebo pomocí cizího softwaru. Naopak v případě multiplayerových her je podvádění považováno za nevhodné a ostatní hráče ve většině případů obtěžuje. Z tohoto důvodu do svých her vývojáři často zahrnují komponenty, které podvádění ve hře více hráčů znemožňují nebo na něj minimálně upozorní zodpovědné osoby.

Dalším problémem, se kterým se toto odvětví potýká, je spolehlivost samotného internetového připojení hráčů a výkon zařízení, na kterém hru hrají. Jednoduše se může stát, že některý z hráčů má příliš velkou odezvu při komunikaci se serverem, což může být zapříčiněno jeho připojením nebo slabým hardwarem, který nestíhá odesílat informace na server. Tyto potíže se mohou ve hře projevat například trhaným, nepřirozeným pohybem hráčova avatara. Nepřátelé tak mohou mít problém "sekajícího se" nepřítele zasáhnout, spojenci si naopak mohou stěžovat na neproduktivního člena týmu.

¹Cheat kódy jsou reprezentovány kombinací kláves nebo vložením kódu přímo do konzole ve hře. Slouží k získání různých výhod ve hře.

3 Herní engine

Cílem této práce je vytvoření trojrozměrné multiplayerové hry. V případě jednotlivce či malých týmu se k vývoji takových videoher používají téměř výhradně herní enginey s již předpřipraveným prostředím pro importování modelů a základními prvky herní logiky a fyziky, mezi něž často patří kolizní zóny objektů a jejich textur, gravitační síla nebo různé druhy osvětlení, a to jak globální nasvětlení celé scény, tak menší zdroje světla. V případě hry pro více hráčů umožňují konfigurovatelný přenos informací mezi klienty hry, případně klienty hry a serverem. Tyto, již hotové komponenty, šetří při vývoji čas a není nutné je implementovat ručně, typicky za použití multimediálních knihoven jako je **DirectX** nebo **OpenGL**. Navíc tato prostředí často zahrnují kompilátory pro různé druhy OS, a tím zajišťují jistou multiplatformnost (viz sekce 2). [4, 3]

3.1 Unity engine

Velkou předností Unity engineu je jeho všestrannost. Unity engine se dá použít k vývoji téměř jakékoli hry, od jednoduché 2D mobilní hry až po velmi komplexní 3D titul pro počítače a herní konzole. Pravděpodobně i díky tomu disponuje tento engine velkou komunitou neustále produkující nejen nové hry, ale i komponenty, které mohou ostatní vývojáři používat. Tyto komponenty jsou označovány jako **assets** a jsou přístupné přímo v editoru Unity engineu, odkud se dají stáhnout zdarma, nebo za poplatek určený autorem. Po stažení se taková komponenta, například model budovy, dá velmi jednoduše importovat do projektu, aniž by vývojář musel opouštět editor. Další výhodou velké komunity Unity engineu je bezesporu dostatek tutoriálů a fór, kde jsou zkušení uživatelé schopni velmi rychle a dobře poradit novým vývojářům s jejich problémy týkajícími se Unity engineu. [5]

Unity engine existuje v neplacené verzi jako Unity **Personal**, která slouží k nekomerčnímu využití a následně ve třech verzích placených – **Plus**, **Pro** a **Enterprise**. Verze Plus obsahuje oproti verzi Personal především více materiálů pro učení se v editoru Unity engineu, možnost vlastní načítací obrazovky ve hře a analytické nástroje pro testování hry. Verze Pro navíc obsahuje rozšířenou podporu pro práci v týmu – větší cloudové úložiště, možnost zakoupení místa pro více jak tři uživatele cloudu, automatické kompilace projektu po nahrání na cloud. Ve verzi Enterprise si uživatel sám zvolí, jaké dodatečné služby k základní verzi Unity chce využívat, a od toho je i odvozena cena. Navzdory tomu, že tento engine má více verzí, je možné v základní bezplatné verzi vytvořit stejnou hru jako v jakékoli jiné placené verzi Unity engineu. [6]

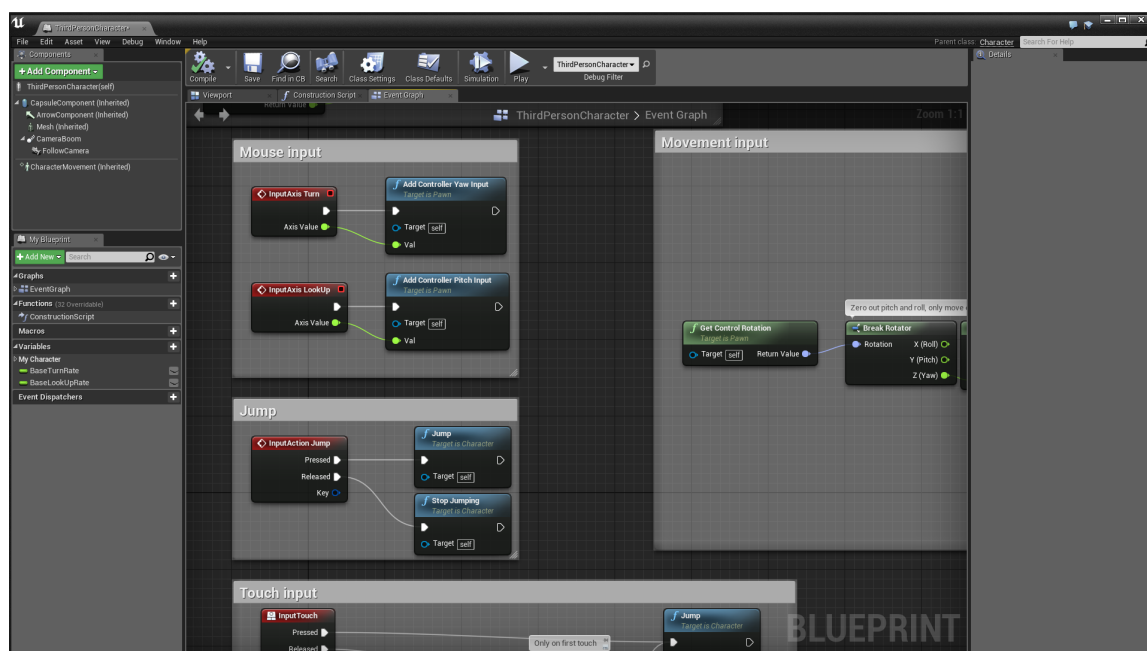
Základními stavebními kameny herní scény jsou takzvané **GameObjecty**. **GameObject** je sám o sobě definován pouze jako bod ve 2D nebo 3D prostoru. Připnutím jednotlivých komponent k takovému bodu docílíme například toho, že se v tomto bodě bude renderovat 3D model (komponenta **Mesh Render**), bude z něj vycházet světlo (komponenta **Light**), bude padat nebo se vznášet (komponenta **Rigidbody**), případně jakákoli kombinace. Důležité komponenty představují skripty, které zajišťují funkčnost hry. Skripty se v Unity engineu píšou v jazyce C# nebo v JavaScriptu, následně jsou připojeny k objektu ve scéně a ovlivňují jeho chování. V Unity

enginu je možné tvořit hry pro následující systémy: Windows, MacOS, Linux Standalone, tvOS, iOS, Lumin, Android, WebGL, PS4, Xbox One. [6]

3.2 Unreal engine

Unreal engine si zakládá na velmi pěkném, někdy až fotorealistickém vykreslování. Není tedy divu, že bývá často používán k vývoji velkých herních titulů jako například *GTA 5*, *Fortnite* nebo *Player Unknown Battlegrounds*. Unreal engine také jako Unity engine podporuje uživateli vytvořené assets, kterých je o něco méně, protože komunita tohoto enginu není tak velká, jak je tomu u Unity enginu. [7, 5]

Nekomerční využití Unreal enginu není nijak zpoplatňováno. Pouze pokud autor vydělá na svém projektu v Unreal enginu více než 3 000 amerických dolarů za čtvrt roku, je povinen za každé takové čtvrtletí zaplatit *Epic Games* 5 % z vydělané částky nad 3 000 amerických dolarů. [7]



Obrázek 1: Unreal engine Blueprints

Typickým rysem Unreal enginu je skriptování logiky hry pomocí takzvaných blueprints. **Blueprint** je dvojrozměrná mapa, kde jsou části kódu reprezentovány a propojeny graficky. Není tedy nutné psát kód. Skriptování je samozřejmě možné i v běžném programovacím jazyce C++. [7, 5]

3.3 CryEngine

CryEngine podobně jako Unreal engine poslouží nejlépe především ke tvorbě AAA herních titulů s velmi realistickým a detailním grafickým zpracováním. Další výhodou CryEnginu je dobrá

Tabulka 1: Srovnání herních enginů [8, 6, 7, 9]

	Unity	Unreal	CryEngine
Podporované platformy	Windows Linux MacOS Android iOS PlayStation 4 Xbox One WebGL tvOS Lumin	Windows Linux MacOS Android iOS PlayStation 4 Xbox One Nintendo Switch AR VR Google Stadia SteamOS HTML5	Windows Linux PlayStation 4 Xbox One Oculus Rift OSVR PSVR HTC Vive
Podporované programovací jazyky	C# JavaScript	C++ Blueprints	C# C++ Lua
Komunita	Velká komunita	Velká komunita	Malá komunita

podpora virtuální reality. CryEngine byl použit při vývoji her jako *Sniper Ghost Warrior 3* nebo *Kingdom Come: Deliverance*. Jelikož je tento engine o něco složitější než ostatní, nedočkal se příliš velké obliby u začínajících vývojářů. CryEngine slouží nejlépe zkušenějším herním společnostem. Použití CryEnginu pro nekomerční účely není zpoplatněno. [8]

3.4 Srovnání herních enginů

V počtu podporovaných cílových platforem jsou na tom nejlépe Unreal engine a hned po něm Unity engine s podporou pro všechny velké OS (Windows, Linux, MacOS), mobilní systémy (Android, iOS) a herní konzole (PlayStation 4, Xbox One). Při vývoji v CryEnginu je nutné počítat s absencí podpory hlavně pro MacOS a mobilní operační systémy.

Jako důležitý faktor při výběru herního enginu se také jeví podporované programovací jazyky pro skriptování herní logiky. Vývojáři, který neprogramuje, nezbývá než volit engine s možností vizuálního skriptování jako jsou blueprints v Unreal enginu. Naopak vývojáři znalí programování mohou využít tradičních programovacích jazyků: C#, C++, JavaScript nebo Lua.

Unreal engine a Unity engine jsou před CryEnginem napřed i ve velikosti vývojářské komunity. Ta je relativně malá kvůli vyšší uživatelské náročnosti prostředí tohoto enginu a kratší době působení na trhu (viz tabulka 1).

4 Návrh multiplayerové hry

Výsledným produktem je akční hra pro více hráčů. Cílem hry je sesbírat víc obilí než protihráči. Zvítězí hráč, který nejrychleji sklídí dané množství obilí. Hráči budou ovládat kombajny vybavené zbraňovým arzenálem. Bude možné ničit ostatní hráče a zdržovat je tak od sběru obilí. Herní mapa bude rozčleněna tak, aby každý hráč začínal odděleně od ostatních a zároveň měl podobné podmínky přístupu k polím s obilím a krytům před nepřátelskou palbou. Design mapy by měl hráče motivovat k interakci s ostatními hráči a k vytváření různých taktik. Síla zbraní, design mapy a další aspekty hry budou doladěny po otestování hry s dobrovolníky.

4.1 Volba softwaru

Pro vývoj této hry byl vybrán Unity engine na základě dobře zpracované obsáhlé dokumentace, velké komunity a podpory více OS. Hra bude dostupná i pro uživatele s méně výkonnými počítači, popřípadě i pro uživatele systémů Linux a iOS, což umožňuje rozšířit okruh potencionálních testerů této práce. [4]

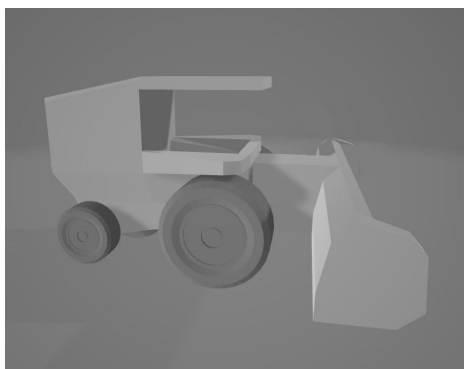
Jako program pro tvorbu a texturování 3D objektů byl zvolen program Blender [10]. Použití Blenderu je zcela zdarma, přičemž nástroje a výsledné modely dosahují stejné kvality jako je tomu u placeného softwaru. 2D grafika bude tvořena v programech Paint.net [11] a Krita [12] částečně za pomoci grafického tabletu. Vzhledem k vysoké časové náročnosti projektu budou pro ušetření času použity i bezplatné modely z *assets store* Unity engine (viz sekce 3.1).

4.2 Návrh síťového propojení

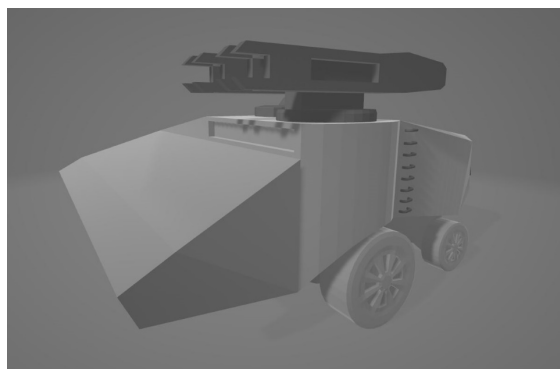
Propojení klientů hry bude řešeno architekturou klient-server. Uživatel, disponující herním klientem, si bude volit mezi připojením k již založené hře a založením vlastní hry. Připojení k cizí hře bude možné na základě zadání veřejné IP adresy počítače hostujícího hráče. Hostující klient (server) bude mít možnost ověřovat platnost zásahů jako opatření proti případným pokusům o podvádění.

5 Vlastní 3D grafika a textury

V tomto projektu se nachází mnoho běžných modelů a textur jako jsou kameny, stromy, tráva, bláto, budovy nebo obloha. Tyto modely jsou zdarma k dispozici jak na internetu, tak v Unity *assets store*, a proto není nutné je pro tento projekt vytvářet znovu. Speciálním případem modelu, který je nutné pro účely tohoto projektu vytvořit a není možné jej sehnat jiným způsobem, je kombajn s dělovou věží. V programu Blender (viz sekce 4.1) byly vymodelovány dvě verze tohoto specifického vozidla. První verze se drží konceptu běžného kombajnu. Verze druhá disponuje více futuristickým designem vycházejícím spíše z konceptu současných bojových vozidel. Vzhledem k už tak nereálnému námětu hry - kombajny bojující o úrodu - byl pro tuto hru vybrán druhý model.



Obrázek 2: První návrh modelu kombajnu

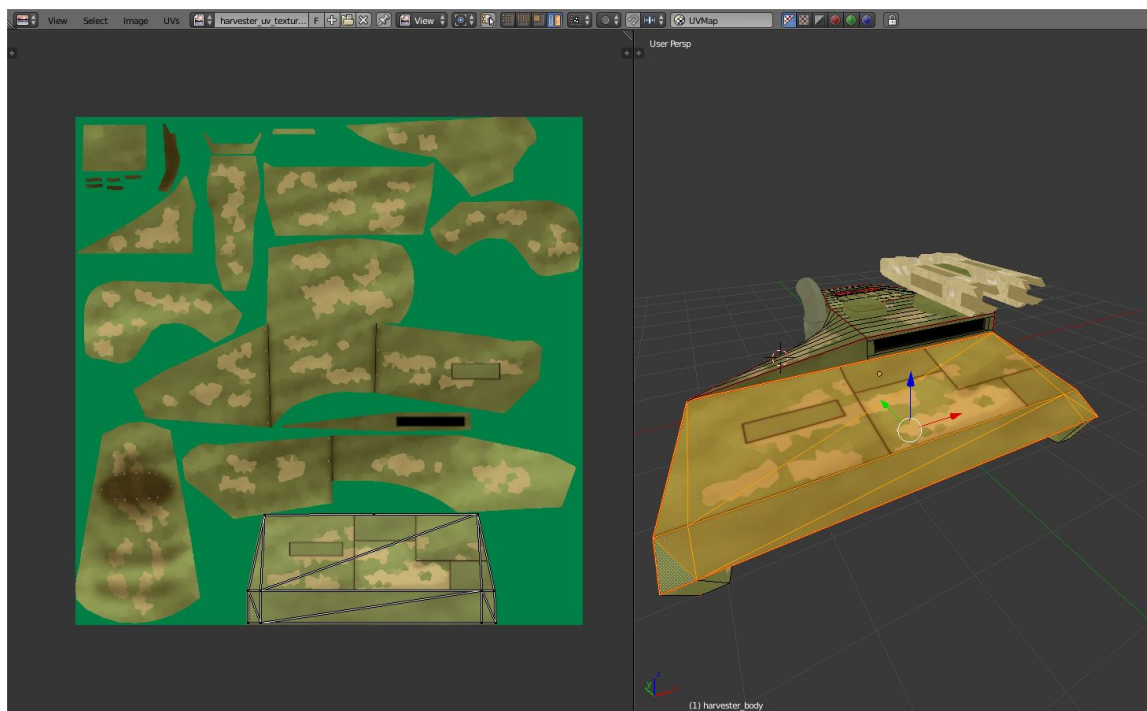


Obrázek 3: Druhý návrh modelu kombajnu

Model kombajnu byl otexturován klasickou metodou **UV mapování** [13] přímo v programu Blender. Při UV mapování je složitější 3D model, složený z mnoha polygonů, rozdělen do menších oblastí, na které se dá snáze nanášet textura v běžném rastrovém grafickém editoru. V tomto případě byl pro editaci textury využit editor Paint.NET.

Textur existuje více druhů. Kromě běžné barevné textury se často využívá i takzvaná normálová mapa. Jedná se o rastrový obrázek, který se aplikuje na model stejně jako textura a způsobuje, že se povrch modelu renderuje ve vyznačených místech plasticky i přes to, že v tomto místě nejsou žádné polygony. V případě modelu kombajnu byla normálová mapa použita na vzorek gumy na kolech a drobné nerovnosti na povrchu karoserie stroje.

Blender používá pro orientaci v 3D prostoru systém souřadnic podle pravidla pravé ruky (*right-handed coordinate system*). To znamená, že osy x a y tvoří pomyslnou podlahu, zatímco osa z určuje výšku. Unity engine používá systém souřadnic podle pravidla levé ruky (*left-handed coordinate system*). Systém souřadnic podle pravidla levé ruky je jakýkoli systém souřadnic, který neodpovídá definici systému souřadnic podle pravé ruky. [14] Konkrétně v Unity engine je tento systém koncipován tak, že osa y reprezentuje výšku. Tento rozdíl způsobuje, že model vyexportovaný v jakémkoli formátu z Blenderu importovaný do Unity engine má špatnou orientaci, což způsobuje řadu problémů. Nejtypičtějším problémem je, že Unity engine po importu



Obrázek 4: UV mapování v programu Blender

modelu z Blenderu považuje spodní část modelu za jeho přední část, čímž znemožňuje použití spousty užitečných vlastností předdefinovaných objektů v Unity engine, které pracují s vektorem směřujícím od GameObjectu (viz sekce 3.1) vpřed (**transform.forward**). Problém byl vyřešen přidáním nového GameObjectu do scény a umístěním modelu kombajnu pod něj tak, aby vektor osy z nového GameObjectu korespondoval s požadovaným směrem pohybu vozidla. K novému GameObjectu jsou následně připínány všechny skripty definující chování vozidla.

6 Implementace multiplayerové hry v Unity engineu

Prvním krokem vývoje byla implementace multiplayeru. Pro testování této komponenty byly založeny dva projekty Unity engineu. Jeden pro klienta hry a druhý sloužící jako server. Přestože jsou aplikace serveru a klienta hry řešeny odděleně, musí se některé parametry jako například délka bufferu pro přenášené informace na straně klienta i serveru shodovat.

6.1 Konfigurace serveru a klienta hry

Síťové propojení je v Unity engineu realizováno skrz statickou třídu **NetworkTransport**. Pro navázání spojení mezi klientem a serverem bylo nutné stanovit port, na kterém budou aplikace komunikovat. Dalším nutným údajem je identifikační číslo serveru získané voláním metody **AddHost** třídy **NetworkTransport**. Nutné je také nastavit maximální možný počet současně připojených hráčů k serveru. Posledním nezbytným atributem je velikost bufferu pro přenášená data. Všechny tyto parametry by se měly shodovat v aplikaci serveru a klienta, pokud jsou řešeny odděleně. Klient navíc vyžaduje IP adresu serveru a musí mít serverem přidělené identifikační číslo spojení získané voláním metody **Connect** třídy **NetworkTransport**.

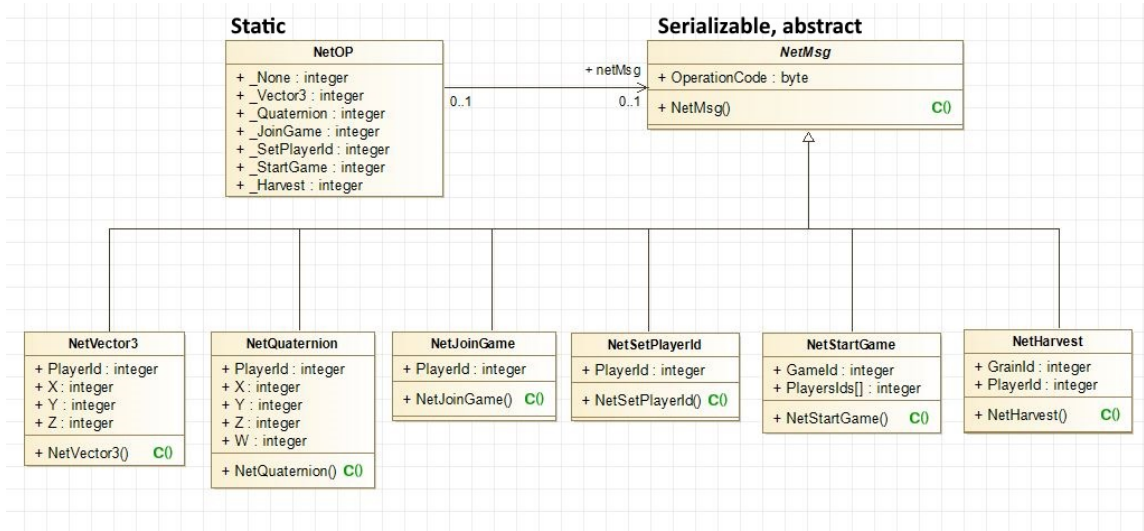
Typicky platí, že čím vyšší kapacitu bufferu nastavíme, tím se prodlouží doba přenosu jednoho packetu. Ideální je tedy optimalizovat aplikaci tak, aby bylo přenášeno co nejméně dat a přitom byla zachována její funkčnost. V případě příliš velkého bufferu Unity engine automaticky před přenosem rozděljuje přenášená data na menší části a po přijetí všech částí je opět sestavuje, což způsobuje značné zpomalení chodu aplikace, a to zvlášť pokud se jedná o multiplayerovou hru běžící v reálném čase.

```
// KONFIGURACE PRO KLIENTA
private const int MAX_USER = 3; // max. počet najednou připojených uživatelů
private const int BYTE_SIZE = 1024; // velikost bufferu
private int hostId; // ID serveru
private int port; // nastaveno uživatelem
private int connectionId; // >= 0
private string serverIp; // nastaveno uživatelem

// KONFIGURACE PRO SERVER
private const int MAX_USER = 3;
private const int BYTE_SIZE = 1024;
private int hostId;
private int port; // nastaveno uživatelem
```

Výpis 1: Ukázka konfigurace klienta hry a serveru

Aby bylo možné data přenášet, je potřeba je **serializovat** na řetězec bitů a následně v cíli **deserializovat**. Jelikož však přenášíme různé typy dat, například pozici hráče, ID zasaženého hráče, rotaci zbraní a tak dále, je nutné tyto typy při přijetí identifikovat a správně deserializovat. Proto byla vytvořena abstraktní serializovatelná třída obsahující ID třídy, jejíž serializovanou instanci přenáší. Z této abstraktní třídy dědí všechny serializovatelné třídy reprezentující různé typy dat. Tak je možné přijatý buffer bitů v cíli identifikovat a přechíst z něj data.



Obrázek 5: Třídní diagram serializovatelných tříd pro přenos herních dat

6.2 Připojení do hry

Poté, co klient úspěšně naváže spojení se serverem, mu server zašle jeho ID spojení. Toto ID je obsaženo ve všech zprávách od klientů. ID je zasíláno přes spolehlivý kanál. **Kanál spolehlivé komunikace** zaručuje doručení paketu do cíle. Doručení je ověřováno a pokud se nepovedlo, paket je odeslán znovu, dokud se doručení nezdaří, nebo není zjištěno, že cílová destinace již neexistuje. Server díky ID rozeznává odesílatele paketu a ukládá si o něm aktuální informace, jako například jeho polohu a zbývající životy.

6.3 Aktualizace pozice vozidel

Data o pohybu hráčova vozidla jsou nepřetržitě odesílána na server. Server po přijetí aktuální polohy hráče odešle tuto informaci ostatním hráčům. Pakety s daty mohou cestovat od klienta k serveru různou rychlostí, a tak je nutné ošetřit s tím spojený problém. Pokud klient odešle sérii paketů s informacemi o poloze, může se stát, že starší informace o poloze se k serveru dostane až poté, co je některá novější informace odeslána zpět klientům ostatních hráčů. Pokud by server tuto starou informaci odeslal klientům, stroje nepřátel by se na dobu, dokud nebude přijat další paket, vrátily do své staré pozice, což způsobuje nepřírozený trhaný pohyb. Tento problém řeší

v Unity enginu komunikace přes **nespolehlivý uspořádaný kanál**. Pakety při komunikaci na tomto kanále obsahují časová razítka, díky nimž je možné ty neaktuální zahazovat. Tento kanál neověřuje doručení paketu do cíle, a tak je komunikace přes něj rychlejší. Nespolehlivý kanál se využívá například i u hlasové komunikace přes internet (VoIP²). Není nutné, aby každý odeslaný paket s pozicí hráče došel do cíle, protože se neustále odesílají nové. Stačí, aby byly přijaty pouze nejnovější pakety.

6.4 Detekce zásahu

Zásahy při střelbě je možno detekovat na základě kolize zasaženého objektu s vystřeleným projektilem. V Unity enginu k tomuto účelu slouží komponenta **Collider**. Pokud je projektil i zasažený objekt vybaven touto komponentou, je možné v kódu na zásah reagovat přepsáním metody **OnCollisionEnter**. Tento způsob detekce zásahu však v případě her pro více hráčů vyžaduje nepřetržitě odesílat informace o poloze letícího projektilu, a to především v případě, kdy je na straně serveru implementována validace zásahu jako ochrana proti podvádění ve hře. Z tohoto důvodu byl v tomto projektu implementován jiný, vhodnější způsob detekce zásahu, který umožňuje validaci zásahu na základě jediného renderu scény z pohledu zbraně střelce.

Ke zbrani hráčova vozidla byla připojena virtuální kamera, která snímá scénu z pohledu hlavně. Při výstřelu je vyrenderován snímek z této kamery, který má nízké rozlišení, aby byl uspořen výpočetní výkon. Za délku strany tohoto čtvercového snímku bylo zvoleno 256 pixelů. Každé vozidlo ve scéně má automaticky přiřazenou unikátní barvu, kterou je při výstřelu dočasně nahrazena originální textura. Barva, kterou obsahuje pixel přesně ve středu tohoto snímku nebo jeho okolí (v případě implementace rozptylu střelby), patří zasaženému objektu. Tímto způsobem je možné identifikovat zasažené vozidlo a jeho ID odeslat v jediném paketu na server. Aby tento způsob detekce zásahu fungoval správně, je nutné nerenderovat objekty, které by mělo být snadné prostřelit. V případě tohoto projektu je takovým objektem tráva, která není jako většina objektů ve scéně reprezentována jako `GameObject`, ale je součástí objektu terénu, který je předpřipravenou komponentou Unity enginu. Skript objektu terénu obsahuje veřejnou booleovskou proměnnou, pomocí které je možné renderování trávy vypnout. Pakety s informací o zásahu se na rozdíl od paketů o poloze odesílají přes spolehlivý kanál, protože je nutné, aby byl každý takový paket doručen.

6.4.1 Vývoj detekce zásahu

Při první implementaci tohoto způsobu detekce zásahu byla přiřazena stejná barva (bílá) každému objektu ve scéně kromě kombajnů s unikátní barvou tak, aby nemohlo dojít k záměně nepřátelského vozidla za statický objekt stejné barvy. Při použití metody měnící materiály na každý objekt ve scéně docházelo při každém výstřelu k prudkému poklesu frekvence snímkování na krátkou dobu (přibližně na dobu jedné sekundy). Pokles byl pravděpodobně způsoben vel-

²Zkratka označuje pojem *Voice over Internet Protocol*.

kým množstvím 3D objektů ve scéně (přibližně 700 objektů) s mnoha polygony. Textury jsou v Unity engineu nanášeny na objekty formou materiálů, které mimo zbarvení 3D modelu definují i jiné grafické vlastnosti, jako například odlesky, emitování světla, plasticitu povrchu a další. Výměna komplexních materiálů na velkém množství složitých 3D modelů při každém výstřelu příliš zatěžovala hardware, což vedlo k dočasnému snížení frekvence snímkování právě aktivní virtuální kamery.



Obrázek 6: První způsob detekce zásahu (snímek z kamery na děle)

Při druhé implementaci byl způsob detekce zásahu zjednodušen tak, že každému vozidlu byla přiřazena netypická barva, jaká by se ve scéně na jiném objektu vyskytovat neměla. Proto nebylo nutné při výstřelu měnit materiály na všech modelech, ale změna se provedla pouze na modelech kombajnů. Tohoto způsobu barevného odlišení se mimo jiné využívá i při volbě barvy **klíčovacího plátna** v oblasti filmové produkce. Typické použití metody **klíčování** (*keying*) spočívá v tom, že se za filmovou scénu umístí barevné plátno. Barva plátna je následně ze snímku odstraněna a pozadí scény je nahrazeno jiným pozadím. Stejně jako v případě barvy vozidel v této implementaci detekce zásahu musí být i barva plátna ve filmové scéně unikátní (viz obrázek 7). [15]



Obrázek 7: Druhý způsob detekce zásahu (snímek z kamery na děle)

Přestože se druhý způsob implementace detekce zásahu ukázal být spolehlivý, bylo po bližším přezkoumání zjištěno, že první způsob implementace detekce zásahů může být realizován tak, aby frekvence snímkování výrazně neklesla. Toho je dosaženo tak, že se metoda měnící materiály na objektu provede v asynchronním režimu. Pro detekci zásahu byla nakonec použita metoda z první implementace s asynchronní metodou pro změnu materiálů na objektu. První způsob implementace byl zvolen z důvodu teoretické vyšší spolehlivosti validace zásahu.

6.4.2 Validace střelby přes okna budov

Speciálním případem objektu, skrz který by mělo být možné střílet jsou okna budov. Modely oken budov v tomto projektu měly v rámci zasazené skleněné průhledné tabulky, které byly při výstřelu s ostatními objekty přebarveny na bílou barvu, a tak ztratily vlastnost průhlednosti. Z toho důvodu nebylo možné zásahy přes okna ověřit. Modely ve scéně, které jsou určeny k přebarvení při výstřelu, jsou označeny štítkem "map". Možným řešením problému by bylo přiřadit těmto skleněným tabulkám jiný štítek a při přebarvování jim nastavit stoprocentní průhlednost. Vzhledem k tomu, že byly tyto skleněné tabulky v oknech sotva postřehnutelné, a modely budov na herní mapě reprezentují neobydlené budovy v dezolátním stavu, byly tyto tabulky zcela jednoduše odstraněny.

6.4.3 Validace zásahu na straně serveru

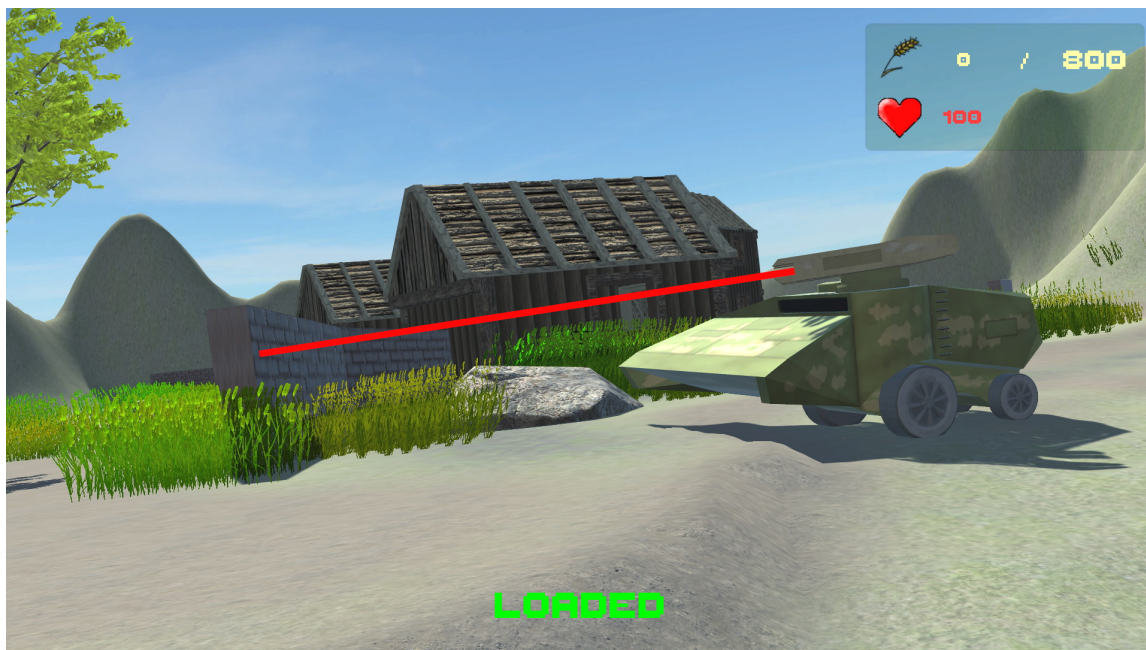
Server si udržuje informace o každém hráči. Tyto informace obsahují náklon a rotaci děla a pozici a rotaci vozidla v herní scéně. Díky těmto informacím je možné při přijetí paketu potvrzujícího zásah sestavit na serveru herní scénu a vyrenderovat snímek z pohledu střílejícího hráče. Tak se ověří platnost zásahu. Díky takovému zabezpečení podvádějícímu hráči nestačí podvrhnout serveru paket s falešnou informací o zásahu, ale musel by zároveň podvrhnout nejnovější pakety o své pozici a pozici ostatních hráčů. Unikátní barva je přidělena objektům ve scéně skriptem hned po spuštění serveru a zůstává na objektech po celou dobu běhu programu.

6.5 Vykreslování laserového paprsku

Při výstřelu je z děla střelce vykreslen laserový paprsek pomocí komponenty **Line Renderer**. Vykreslený paprsek však nemá kolizní zóny. Vykresluje se od jednoho bodu ve 3D prostoru ke druhému. Z toho důvodu je problém určit, v jakém bodě paprsek ukončit, chceme-li aby se zastavoval o pevné překážky. Pro prototyp hry byl určen jako koncový bod paprsku bod v konstantní vzdálenosti od děla. Z toho důvodu paprsek často při výstřelu protnul několik objektů o které by se měl správně zastavit.

Prvním pokusem o implementaci v tomto projektu bylo řešení, kdy byl z děla střelce vystřelen projektil velkou rychlostí a místo, kde projektil narazil na překážku bylo určeno jako konečný bod paprsku. Tato metoda se však v několika směrech ukázala jako nevhodná. Prvním problémem takové implementace byla prodleva mezi výstřelem a dosažením první překážky pomocí

projektilu, která způsobovala že se paprsek vyrenderoval až za několik desetin vteřiny po výstřelu. Větší rychlost projektilu způsobovala to, že kolizní zóny na modelu projektilu přestávaly zaznamenávat kolize s ostatními objekty ve scéně. Další problém se projevoval tím, že projektil při nárazu pohnul s objekty s kolizními zónami a komponentou Rigidbody.



Obrázek 8: Zastavení laserového paprsku o překážku

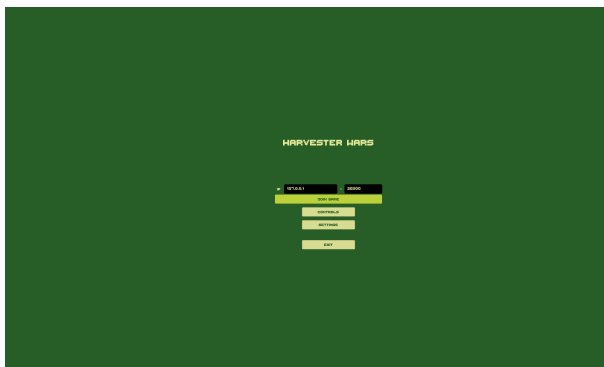
Z těchto důvodů byl zvolen jiný způsob implementace, pro který byla použita funkce **Raycast**, která v požadovaném směru definovaným vektorem zjistí nejbližší GameObject s aktivními kolizními zónami. Funkce Raycast objektu **Physics** jako jeden ze svých výstupních parametrů vrací objekt **RaycastHit**, který mimo jiné obsahuje i informaci o přesném místě kolize Raycastu s GameObjectem. Tento bod v prostoru byl použit jako koncový bod paprsku laseru.

6.6 Uživatelské rozhraní hry

Uživatelské rozhraní, dále jen **UI** (*user interface*), klienta hry je tvořeno hlavní nabídkou, ze které se může hráč po zadání IP adresy a portu serveru připojit rovnou do hry. Ve hře je UI tvořeno počítadlem životů a sesbírané úrody v pravém horním rohu obrazovky a indikátorem přebíjení děla ve spodní středové části obrazovky.

Zajímavým prvkem UI ve hře je také vyskakovací nabídka při stisknutí klávesy *escape*, která je realizována jako samostatná scéna obsahující pouze plátno s prvky UI (texty, tlačítka, vstupy pro text, panely). Díky tomu, že se jedná o samostatnou scénu, je možné ji spustit jako novou vrstvu nad spuštěnou hrou a případně ji opět jediným příkazem

```
SceneManager.UnloadSceneAsync(sceneNumber)
```

Obrázek 9: Hlavní nabídka



Obrázek 10: Vyskakovací okno ve hře

ukončit. Tato nabídka hráče naviguje buď k předčasnému ukončení hry, nebo k návratu do hry.

Problémovým prvkem UI byla obrazovka shrnující výsledek hry. Po ukončení hry jsou výsledky uloženy do **PlayerPrefs**, což je objekt, do kterého se ukládají hodnoty pod zvoleným klíčem. Problém nastal při testování, kdy bylo zjištěno, že hodnoty uložené do PlayerPrefs se ukládají do jediného souboru na disku. Při testování více klientů hry na jediném zařízení si klienti soubor vzájemně přepisují, a tak jsou jejich herní výsledky vždy shodné. Hra není určena k tomu, aby byla vícekrát souběžně spuštěna na jediném zařízení a PlayerPrefs jsou podle dokumentace Unity engine nejlepší způsobem přenosu dat mezi herními scénami. Následující chyba tedy nebyla opravena, ale bylo přidáno varování do hlavní nabídky hry, které se zobrazuje, pokud je na jednom zařízení spuštěno více instancí hry.

6.7 Ostatní herní mechaniky

Dále byl implementován sběr obilí, za který se hráči přičítá skóre. Hráč, který nejrychleji sesbírá dané množství obilí, vyhrává. Hráči se mohou přepnout do alternativního (ostřelovacího) režimu střelby, ve kterém vidí herní scénu z pohledu děla a mohou přibližovat a oddalovat obraz kolečkem myši podobně jako tomu je u dalekohledu.

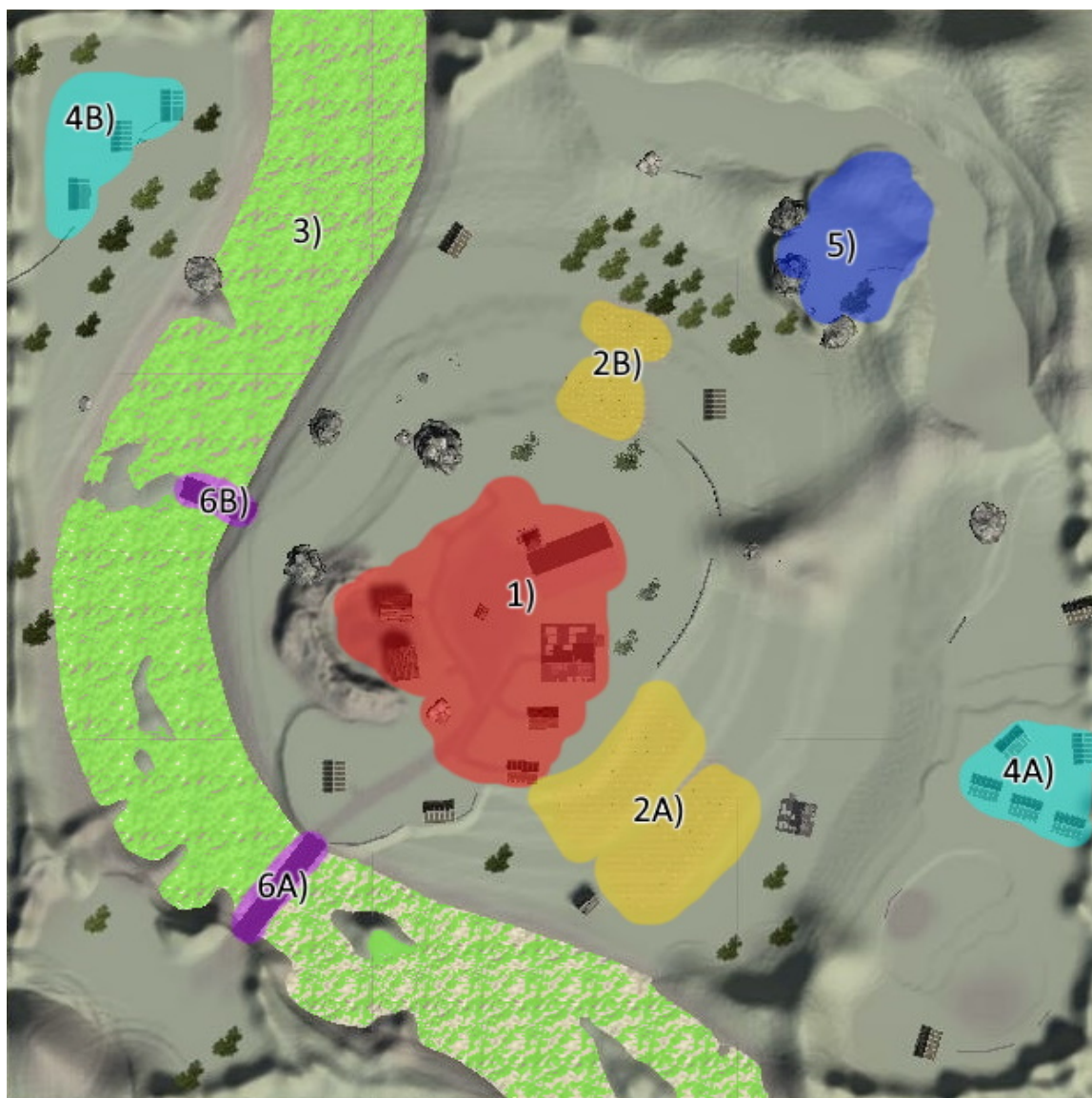


Obrázek 11: Běžný režim střelby



Obrázek 12: Ostřelovací režim střelby

6.8 Design herní mapy



Obrázek 13: Mapa oblastí

Herní mapa je rozčleněna do několika specifických oblastí, což má hráči umožňovat lepší orientaci a celkově působit víc realistickým dojmem. Tyto oblasti jsou: (1) vesnice s kostelem ve středu mapy ležící na mírném kopci, (2) dvě pole na stráni kopce, (3) potok pod kopcem, (4) ostatní budovy, (5) skála a (6) dva mosty přes potok. Součástí mapy je mnoho terénních nerovností a jiných překážek, za které se hráči mohou schovávat před nepřátelskou palbou a za nimiž mohou například nepozorovaně vniknout nepříteli do zad. Zda těchto aspektů mapy skutečně hráči využívají, bude zjištěno při testování. Na základě nových poznatků bude herní mapa upravena. Na hranice herní mapy a na příliš příkré terénní nerovnosti byly umístěny neviditelné kolizní zóny, aby hráči nemohli omylem opustit herní mapu nebo se s vozidlem

dostat do míst, kde je to nežádoucí. Tyto kolizní zóny jsou však důkladně zamaskovány terénem a jinými překážkami, aby hráč nebyl přímo konfrontován s "neviditelnou zdí".

7 Testování hry

Za účelem odhalení případných nedostatků bude provedeno testování. Dílčím cílem testování bude zjistit, zdali hráči využívají implementovaných herních mechanik a možností herní mapy. Na základě výsledků testování budou na hře provedeny úpravy.

7.1 Metodika

Pro testování budou vybráni pouze lidé, kteří v současnosti aktivně hrají hry a vlastní dostatečně výkonný hardware (viz příloha B), protože se zjevně jedná o cílovou skupinu. Zdali je tester aktivním hráčem bude posouzeno na základě nahraných hodin na herních titulech z let 2016 až do současnosti a stále aktivních MMO titulech (viz sekce 2). Limitem bude 200 hodin a víc herního času v součtu. Respondenti doloží svůj herní čas snímkem nebo snímkem obrazovky s nahraným časem.

Testování bude probíhat ve třech fázích. V první fázi si skupina hráčů hru v několika nemonitorovaných hrách vyzkouší. Následně bude odehráno pět monitorovaných her. Ve třetí fázi bude každému hráči formou strukturovaného rozhovoru položeno několik otázek. Odpovědi na otázky budou zaznamenány do tabulek (viz tabulky 2, 3 a 4), které budou umístěny do příložených souborů. Testování by se mělo účastnit šest hráčů. Proběhnou tři vlny testování. Po každé vlně testování budou provedeny úpravy na základě nejčastěji zmiňovaných nedostatků při strukturovaných rozhovorech.

Pohyb hráčů ve hře bude serverem zaznamenáván do textového souboru ve formátu **csv** (formát textového souboru, ve kterém jsou jednotlivé informace od sebe odděleny čárkou) [16] a následně externím programem zakreslen barevně do obrázku herní mapy. Externím programem bude skript v jazyce Python [17]. Díky monitorování hry bude zjištěno, které části herní mapy jsou nejvíce využívány. Nevyužité části, pokud se nějaké vyskytnou, budou přetvořeny tak, aby byly pro hráče atraktivnější. Vzhledem k velikosti herní mapy se předpokládá, že hráči budou potřebovat minimálně dvě hry na to, aby hru poznali a zjistili výhodné pozice a strategie. Na základě tohoto předpokladu by měli hráči v prvních hrách zkoumat své možnosti. Až poslední hry, kde hráči budou implementovat své strategie na základě zkušeností, budou obsahovat validní informace o využití herní mapy.

7.2 První vlna testování

Z provedených rozhovorů s testery hry vyplývá, že nejzávažnějším problémem, na který v průběhu hraní všichni narazili, je chyba, která prakticky znemožňuje hraní a která nastává při druhé hře, pokud klient hry nebyl restartován. Chyba se projevuje tak, že je možné střilet i v čase přebíjení a žádné zásahy se nezapočítávají. Také některá nepřátelská vozidla mohou změnit barvu textury na barvu používanou pro detekci zásahu. Jiné podobně závažné chyby nebyly odhaleny.

Všem testerům vadil směr otáčení vozidla při couvání. Kombajny ve hře se ovládají pro počítačové hry typickým způsobem pomocí kláves W, A, S a D, přičemž klávesa W pohybuje vozidlem dopředu, klávesa S dozadu, klávesa A doleva a klávesa D doprava. Při couvání však klávesa A pohybuje vozidlem doprava a klávesa D doleva. Testeré také doporučovali přidat větší množství zbraní, popřípadě bonusy, které by se mohly nacházet na herní mapě a bylo by možné je sebrat. Bylo navrženo, aby bylo zničené vozidlo před opětovným vypuštěním do hry na nějaký čas pozdrženo, aby se hráčům vyplatilo střílet po nepřátelských vozidlech. Dalším diskutovaným tématem při rozhovorech byla možnost přidání oprav vozidel do hry. Zajímavým poznatkem je, že přestože testeré shledávají hru zábavnou, hráče nic nemotivuje k dlouhodobějšímu hraní. Tento problém by mohlo vyřešit odměňování hráčů zobrazováním více detailů o proběhlé hře a také udělování ocenění za nadstandardní výkony ve hře.



Obrázek 14: První hra



Obrázek 15: Druhá hra



Obrázek 16: Třetí hra



Obrázek 17: Čtvrtá hra



Obrázek 18: Pátá hra

Data o pohybu hráčů, nasbíraná v průběhu her, poukazují na to, že se hráči pohybují především v jižní části herní mapy okolo jižního pole. Během rozhovorů několik testerů poukázalo

na to, že severně položené pole může být problematické, jelikož se hráči velice často dostávají do konfliktu v jižní části mapy. Pokud se jeden z hráčů z takového konfliktu nepozorovaně vytratí, a nebo se mu záměrně vyhne, má pak dobrou šanci sesbírat nerušeně požadované množství úrody na poli severním.

Testeři při hraní využili všechny oblasti herní mapy, až na dvě. Skupina budov v severozápadní části mapy nebyla využita pravděpodobně kvůli své odlehlosti od obou polí. Kopec nad vesnicí v severovýchodní části mapy byl zamýšlen jako výhodná střelecká pozice. V průběhu testování se však ukázalo, že střelba do nepřátelských vozidel je v současném stavu hry efektivní pouze tehdy, pokud při ní střelec samotný těží obilí.

7.3 Úprava hry na základě první vlny testování

Chyba, která způsobovala problémy ve druhé hře spuštěné v jedné instanci aplikace klienta, byla způsobena tím, že Unity engine nemaže obsah statických proměnných při ukončení scény. Takto docházelo ke znovunačtení starých statických dat při spuštění nové hry. Problém se týkal skriptu **client.cs**, který zajišťuje síťovou komunikaci a uchovává statická data s informacemi o hráči, ostatních hráčích a pozicích základen, kde jsou vozidla vysazena při začátku hry, nebo pokud jsou zničena. Chyba byla opravena odstraněním statického charakteru těchto proměnných. Přístup k těmto proměnným z ostatních skriptů je zajištěn vyhledáním objektu **client** pomocí metody **Find** Unity objektu **GameObject** (viz sekce 3.1) a načtením jeho stejnojmenné komponenty - instance skriptu **client.cs** (**GetComponent<Client>()**).

```
private Client client;

private void Start()
{
    client = GameObject.Find("Client").GetComponent<Client>();
    client.GamePlayer.HealthPoints = 100;
}
```

Výpis 2: Řešení problému se statickými proměnnými

Dále byl změněn směr otáčení vozidla při couvání tak, že klávesa A otáčí vozidlo i při couvání doleva a klávesa D naopak doprava. Dále byly provedeny následující změny: (1) přidání opravy vozidel ve formě sesbíratelných lékárníček rozmístěných po mapě, (2) zdržení hráče na základně po dobu 10 sekund, kam se vrátí, pokud je jeho vozidlo zničeno, (3) vytvoření bonusu zrychlujícího střelbu (rozmístěn po mapě stejně jako lékárníčky), (4) odebrání pole na severu mapy a (5) rozdělení pole na jihu překážkami na dvě části. Zaměřovač je nyní viditelný i mimo ostřelovací režim. Hráči nově naleznou informace o ovládání hry v menu.

Na základě stížností hráčů na málo notifikací o zásahu byl přidán zvuk, který se přehraje po obdržení zásahu. Zároveň se při obdržení zásahu na obrazovce na 0,8 sekundy zobrazí skvrny,

aby bylo možné zásah zaregistrovat i bez zvuku. Pokud hráči klesnou životy pod 30 (maximum je 100 životů), spustí se z motoru dým tvořený částicovým efektem (*particle effect*). Částicový efekt je v Unity enginu reprezentován objektem, který podle požadovaného nastavení generuje proud částic. Tyto částice jsou 2D obrázky, které jsou v prostoru 3D scény namapovány na čtvercový polygon. V případě tohoto projektu se jedná o obrázek obláčku dýmu.

7.4 Druhá vlna testování

Výsledky testování druhé verze hry ukazují, že bylo dosaženo výrazného zlepšení hratelnosti. Při druhé vlně testování se nevyskytly žádné závažné chyby, které by hráči znemožňovaly hru hrát. Na rozdíl od první vlny testování se testeři neshodují na žádném nedostatku hry. Naopak se všichni tři testeři shodují na tom, že herní mapa je v pořádku a není třeba na ní už nic upravovat a také na tom, že nová verze hry je lepší než ta předchozí.

Zmíněné nedostatky, které budou opravovány jsou: (1) možná absence notifikace o zásahu v některých případech, (2) obdržení poškození při čekání na základně, (3) zpožděný zvuk výstřelu, (4) orientace hráče na severní základně, (5) ukládání poslední použité IP adresy serveru a (6) zavírání menu ve hře klávesou *escape*.



Obrázek 19: Šestá hra



Obrázek 20: Sedmá hra



Obrázek 21: Osmá hra

Na schématech bitev z druhé vlny testování je vidět, že boje probíhaly konzistentně o jediné pole s obilím na mapě. Hráči v průběhu bitev využívali bonusy a překážky rozmístěné po mapě ke zvýšení své taktické výhody. Bonus přidávající životy je vyznačen v mapách červeným křížem a bonus zvyšující rychlost přebíjení je vyznačen žlutým bleskem.

7.5 Úprava hry na základě druhé vlny testování

Absence notifikací o zásahu vozidla se neprokázala. V případě chyby, která se měla projevit poškozením vozidla při zásahu v době, kdy vozidlo čekalo na základně na opětovné nasazení, se ukázalo, že k takovému poškození nemůže dojít. Pokud je hráč v této situaci zasažen, ukáží se notifikace oznamující zásah, životnost vozidla však není ovlivněna. Tak nejspíš došlo k tomu, že se jeden respondent domníval, že jeho vozidlo jiný tester ničí v situaci, kdy se nemůže bránit. Do



Obrázek 22: Devátá hra



Obrázek 23: Desátá hra

okolí základen bylo umístěno více překážek, aby nebylo možné útočit na nově nasazená vozidla (což je nežádoucí jev označovaný jako base camping).

Dále byla opravena počáteční orientace hráče na severní základně tak, aby hráčovo vozidlo začínalo hru jako ostatní vozidla, orientováno čelem k poli, kde se odehrává většina střetnutí mezi hráči. Bylo přidáno ukládání poslední použité IP adresy serveru do PlayerPrefs. Menu ve hře je nyní možné zavřít klávesou *escape*, stejně jako otevřít.



Obrázek 24: Rozptyl při střelbě

Respondenti kladně hodnotili dynamiku soubojů oproti předešlé verzi hry, a to obzvláště boj zblízka. Jeden z respondentů navrhoval implementaci stabilizátoru děla, které by umožňovalo v takových situacích lepší zaměřování. Jiný respondent navrhoval řešení, kdy by bylo možné nepřítele při blízkém kontaktu poškodit nárazem. Obě tyto mechaniky by však tyto souboje spíše urychlily, a proto byl do hry přidán rozptyl při střelbě, pokud se vozidlo pohybuje. Poloměr rozptylu byl stanoven na 15 pixelů od středu 256 pixelového čtvercového snímku z kamery na děle. Tímto způsobem by mohlo být dosaženo toho, že souboje za pohybu na krátkou vzdálenost budou delší a zajímavější, jelikož bude těžší nepřítele zasáhnout. Zároveň by díky této úpravě mohly hrát ještě větší roli bonusy zvyšující rychlost přebíjení děla, které poskytnou více pokusů k zasažení cíle za krátký čas. Jako notifikace o vyšším rozptylu slouží záměrný kříž, který se rozšíří a zčervená, pokud se vozidlo pohybuje.

7.6 Třetí vlna testování

Třetí vlny testování se zúčastnili jiní respondenti než předchozích dvou vln. Závěry druhé skupiny respondentů zjištěné při rozhovorech se v mnoha bodech shodovaly se závěry předchozí skupiny. Respondenti v obou skupinách se shodují, že by se měla zvýšit rychlost couvání a mělo by být přidáno více druhů zbraní, statistiky na konci hry a bonusy (například štít pro vozidla). Respondenti ve druhé skupině se navíc shodují že by mělo být obilí více rozptýleno po herní mapě a mělo by ho být větší množství. V předchozích testováních se však ukázalo, že pokud je obilí více rozptýleno po mapě, tak spolu hráči nejsou dostatečně konfrontováni. Jako možné řešení se nabízí rozšířit stávající pole tak, aby byli hráči stále situováni do stejné části mapy a zároveň bylo k dispozici víc míst s obilím.



Obrázek 25: Jedenáctá hra



Obrázek 26: Dvanáctá hra



Obrázek 27: Třináctá hra

Druhá skupina testerů využívala možnosti herní mapy podobně jako předchozí skupina. Hráči sbírali bonusy rozmístěné po mapě a pohybovali se po poli nebo kolem něj. Díky přidávanému rozptylu zbraní probíhaly souboje mezi hráči podle očekávání déle a hráči již nepožadovali žádné úpravy na chování děl kombajnů, nebo poškození nárazem.



Obrázek 28: Čtrnáctá hra



Obrázek 29: Patnáctá hra

8 Závěr

Cílem práce bylo vytvořit multiplayerovou počítačovou hru. Hra měla vzniknout na základě seznámení se s problematikou vývoje počítačových her a následných testování s hráči. Výsledkem je plně funkční multiplayerová hra pro tři hráče (viz příloha A), která se dá hrát v lokální síti, nebo ve veřejné síti. Ve veřejné síti se dá hra hrát, pokud má hráč hostující server od svého poskytovatele internetového připojení přidělenou veřejnou IP a uzpůsobí komunikaci serveru s klienty svou domácí síťovou topologií. Jako součást práce vznikl manuál pro vytvoření vlastní virtuální lokální sítě (viz příloha D).

Součástí vývoje bylo i vytvoření části vlastního grafického obsahu. Vlastním grafickým obsahem je model kombajnu i s texturami, model kupy obilí, které je kombajny sbíráno, ikony uživatelského rozhraní a modely sesbíratelných vylepšení. K jejich vytvoření byly použity programy Blender, Krita a Paint.net.

Testy s hráči proběhly v pořádku, bez větších komplikací. Aplikace serveru v průběhu testování sbírala data o pohybu hráčů. Takto získané souřadnice byly poté externím programem vizualizovány do podoby mapy s barevně vyznačenými trasami pohybu hráčů. Testeři byli navíc podrobeni strukturovanému rozhovoru. Nasbíraná data výrazně pomohla vylepšit hratelnost výsledné hry, což bylo zjištěno hned při druhé vlně testování.

Jak při samotné implementaci hry, tak i při testování s hráči, byly odhaleny chyby různého charakteru, které se podařilo napravit. Mezi tyto chyby patřily například statické proměnné držící si svou přidělenou hodnotu i při opětovném načtení scény nebo problém při importu modelů do Unity engineu způsobený rozdílnými souřadnicovými systémy.

Při dalším rozšiřování hry by měl být implementován informační systém, který by ukládal výsledky hráčů v jednotlivých hrách a umožňoval by například vytvoření žebříčku hráčů podle úspěšnosti, což by motivovalo hráče k dlouhodobějšímu hraní. Pro provoz takového systému by také bylo ideální hostovat aplikaci serveru na vlastních serverech. Tyto závěry vyplývají z dat, získaných od testerů hry.

Literatura

1. NANDHAKUMAR, Joe; PANOURGIAS, Nikiforos S.; SCARBROUGH, Harry. From Knowing It to "Getting It": Envisioning Practices in Computer Games Development. *Information Systems Research* [online]. 2013, roč. 24, č. 4, s. 933–955 [cit. 2020-03-11]. ISSN 10477047. Dostupné z DOI: 10.1287/isre.2013.0482.
2. FREEMAN, Guo; MCNEESE, Nathan J. Exploring Indie Game Development: Team Practices and Social Experiences in A Creativity-Centric Technology Community. *Computer Supported Cooperative Work: The Journal of Collaborative Computing* [online]. 2019, roč. 28, č. 3/4, s. 723–748 [cit. 2020-03-11]. ISSN 09259724. Dostupné z DOI: 10.1007/s10606-019-09348-x.
3. B.AHMAD, Norita; BARAKJI, Salahudin Abdul Rahman; SHAHADA, Tarak Mohamed Abou; ANABTAWIB, Zeid Ayman. How to launch a successful video game: A framework. *Entertainment Computing* [online]. 2017, roč. 2017, č. 23, s. 11 [cit. 2020-02-20]. Dostupné z DOI: <https://doi.org/10.1016/j.entcom.2017.08.001>.
4. MAHER, Kathleen. Game Engines Power Content Creation. *Computer Graphics World* [online]. 2019, roč. 42, č. 1, s. 58–62 [cit. 2020-03-11]. ISSN 02714159. Dostupné z: <http://search.ebscohost.com/login.aspx?direct=true%5C&db=asn%5C&an=137900173%5C&scope=site>.
5. PETTY, Josh. Top 12 Free Game Engines For Beginners & Experts Alike. *Concept Art Empire* [online]. 2020 [cit. 2020-02-25]. Dostupné z: <https://conceptartempire.com/free-game-engines/>.
6. *Unity engine* [online]. U.S.: Unity Technologies, 2020 [cit. 2020-02-25]. Dostupné z: <https://unity.com/>.
7. *Unreal engine* [online]. U.S.: Epic Games, 2020 [cit. 2020-02-25]. Dostupné z: <https://www.unrealengine.com/>.
8. *CryEngine* [online]. Crytek GmbH., 2020 [cit. 2020-03-11]. Dostupné z: <https://www.cryengine.com/>.
9. CHRISTOPOULOU, Eleftheria; XINO GALOS, Stelios. Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices. *International Journal of Serious Games*. 2017-12, roč. 4, s. 21–36. Dostupné z DOI: 10.17083/ijsg.v4i4.194.
10. *Blender* [online]. USA: Blender Foundation, 2020 [cit. 2020-03-11]. Dostupné z: <https://www.blender.org/>.
11. *Paint.net* [online]. DotPDN LLC, 2020 [cit. 2020-03-11]. Dostupné z: <https://www.getpaint.net/>.
12. *Krita* [online]. Krita Foundation, 2020 [cit. 2020-03-11]. Dostupné z: <https://krita.org/>.

13. *UV Editor: Introduction* [online]. Blender, 2020 [cit. 2020-03-19]. Dostupné z: <https://docs.blender.org/manual/en/latest/editors/uv/introduction.html>.
14. *Left-Handed Coordinate System* [online]. Wolfram Research, Inc., 1999-2020 [cit. 2020-03-19]. Dostupné z: <https://mathworld.wolfram.com/Right-HandedCoordinateSystem.html>.
15. *What Is Chroma Key?* [online]. Australia, 34 Park Street, South Melbourne: Lumeo, 2020 [cit. 2020-03-17]. Dostupné z: <https://www.lumeo.com.au/blog/chroma-key>.
16. *Common Format and MIME Type for CSV Files* [online]. SolidMatrix Technologies, Inc., 2005 [cit. 2020-03-22]. Dostupné z: <https://www.ietf.org/rfc/rfc4180.txt%5C#page-1>.
17. *Python* [online]. Python Software Foundation, 2001-2020 [cit. 2020-03-22]. Dostupné z: <https://www.python.org/>.

A Přílohy

V přiložených souborech se nalézají spustitelné soubory serveru (**server_win** a **server_linux**) a klienta hry (**client_win** a **client_linux**) pro operační systémy Linux a Windows. Dále pak přiložené soubory obsahují projekty Unity engine pro klienta (**Unity projekt - Client**) a server (**Unity projekt - Server**) hry. Pro vývoj hry byla použita verze **Unity 2018.3.8f1**. Pro procházení nebo editování skriptů může být použito libovolné C# IDE podporované editorem Unity engine (pro vývoj bylo použito Microsoft Visual Studio 2017).

B Hardwarové požadavky

Minimální HW požadavky byly získány z dokumentace Unity engine. [6] Optimální HW požadavky byly zjištěny podle jednoho z testerů s nejslabším HW.

B.1 Optimální hardwarové požadavky

- OS: x64 Windows 10
- CPU: AMD Ryzen 5 3500U a lepší
- RAM: 4GB DDR4 a víc
- GPU: AMD Radeon Vega 8 a lepší
- Místo na disku: 1GB

B.2 Minimální hardwarové požadavky

- CPU: x86, x64 architecture with SSE2 instruction set support
- RAM: 4GB DDR3
- GPU: DX10, DX11, DX12 capable
- Místo na disku: 1GB

C Seznam použitého softwaru

- Unity 2018.3.8f1
- Microsoft Visual Studio 2017
- Paint.NET 4.2
- Krita

D Návod ke zprovoznění virtuální LAN

Pomocí virtuální lokální sítě může být nahrazeno zdlouhavé nastavování síťových prvků na straně hráče provozujícího server. Jako rychlé, bezproblémové řešení se nabízí použití programu **Hamachi**, který je zdarma ke stažení na adrese <https://www.vpn.net/>. Aby bylo možné program používat, je nutné se na webu **vpn.net** registrovat.

Poté, co se všichni hráči zaregistrují a nainstalují si program Hamachi, vytvoří provozovatel serveru novou síť a její jméno a heslo zpřístupní ostatním hráčům. Hráči se připojí do vytvořené virtuální LAN. Následně hráč provozující server spustí aplikaci serveru. Ostatní hráči použijí IP adresu hráče se serverem ve virtuální síti jako adresu serveru.

E Výsledky rozhovorů s testery po testování hry

Tabulka 2: Výsledky rozhovorů v první vlně testování

	Tester 1 (Daniel)	Tester 2 (Václav)	Tester 3 (Milan)
Odhalili jste ve hře nějaké bugy (chyby)? Pokud ano, popište jaké.	Při spuštění další hry, pokud není herní klient restartován, se nezapočítávají zásahy a textury na strojích mají jinou barvu.	Pokud se herní klient nerestartuje, tak se ve hře objevují bugy.	Pokud se herní klient nerestartuje, tak se ve hře objevují bugy.
Sekala se vám hra? Pokud ano, v jaké situaci?	Pohyb nepřátelských vozidel byl lehce trhaný, pokud byla vozidla v dálece.	Nic nezaznamenal.	Nic nezaznamenal.
Výhovovalo vám ovládání vozidla? Co byste změnili?	Při couvání a zatačení zároveň se vozidlo otáčí jiným směrem než by hráč čekal.	Při couvání a zatačení zároveň se vozidlo otáčí jiným směrem než by hráč čekal.	Při couvání a zatačení zároveň se vozidlo otáčí jiným směrem než by hráč čekal.
Výhovovaly vám zbraně? Co byste změnili?	Velká citlivost myši v odstřelovacím režimu pokud je dalekohled hodně přiblížený. Přidat zvuk, když je vozidlo hráče zasaženo. Přidat víc možností zbraní (typy munice, režimy střelby) Ukazovat celkové poškození udělené nepřátelským vozidlům v závěrečném shrnutí výsledků hry.	Přidat zaměřovač při střelbě za jízdy.	Přidat různé druhy zbraní nebo munice.
Jak hodnotíte herní mapu a co byste případně změnili na jejím rozložení?	V pořádku.	Pole na severu mapy umožňuje jednomu z hráčů snadno zvítězit, pokud se ostatní dva perou o východní pole. Vytvořit víc menších polí.	Odebrat severní pole, přidat víc hráčů, nebo mapu zmenšit.
Byl průběh hry dostatečně plynulý? Narušovalo něco váš požitek ze hry?	Nic nenarušovalo herní zážitek.	Střelba za jízdy bez zaměřovače.	Nic nenarušovalo herní zážitek.
Existuje nějaká herní mechanika, kterou byste rádi ve hře viděli? Co můžeme zlepšit?	Ocenění za akce ve hře (za skok z mostu, přetočení vozidla na střechu) Doplnění životů, třeba ve stodole (tematicky) Přidat příručku s ovládáním hry.	Sbírání vylepsení (rychlost vozidla, zrychlené přiblížení, doplnění životů) Pokud je hráč zničen, zdržet jej několik vteřin než bude moci znovu hrát.	Po zničení nepřítele připsat hráči jeho natežené body.

Tabulka 3: Výsledky rozhovorů ve druhé vlně testování

	Tester 1 (Daniel)	Tester 2 (Václav)	Tester 3 (Milan)
Odhalili jste ve hře nějaké bugy (chyby)? Pokud ano, popište jaké.	V některých případech se možná neobjevovaly skvrny na obrazovce při zásahu. (není si jistý)	Obdržel poškození při čekání na spawnu.	Neodhalil.
Sekala se vám hra? Pokud ano, v jaké situaci.	Nesekala.	Nesekala.	Pohyb nepřátel je někdy lehce trhaný.
Vyhovovalo vám ovládání vozidla? Co byste změnili?	Couvání je pomalé.	V pořádku.	V pořádku.
Vyhovovaly vám zbraně? Co byste změnili?	Vylepšení zrychlující střelbu je dobré. Nepřidávat víc vylepšení (pouze v případě, že by byly přidány do jiných hratelných map) Velká citlivost v ostřelovacím režimu.	Ve hře je důležitější sbírat obilí než střílet - nic neměnit.	Vyhovovaly.
Jak hodnotíte herní mapu a co byste případně změnili na jejím rozložení?	V pořádku.	V pořádku.	V pořádku.
Byl průběh hry dostatečně plynulý? Narušovalo něco váš požitek ze hry?	Zasekl se o kámen a kvůli tomu byl zničen jiným hráčem.	Zvuk výstřelu přichází se zpožděním.	Na severním spawnu hráči začínají orientovaní čelem od bojiště.
Existuje nějaká herní mechanika, kterou byste rádi ve hře viděli? Co můžeme zlepšit?	Zavírat menu stejně jako se otevírá - klávesa ESC. Zvážit přidání stabilizátoru děla.	Poškození při nárazu vozidla do vozidla sběrací radlicí.	Ukládání poslední použité IP adresy serveru.
Jak hodnotíte tuto verzi hry (v porovnání s předchozí)?	Tato nová verze je lepší.	Tato nová verze je lepší.	Tato nová verze je lepší.

Tabulka 4: Výsledky rozhovorů ve třetí vlně testování

	Tester 4 (Radovan)	Tester 5 (Tomáš)	Tester 6 (Stefan)
Odhali jste ve hře nějaké bugy (chyby)? Pokud ano, popište jaké.	Neodhalil.	Neodhalil.	Neodhalil.
Sekala se vám hra? Pokud ano, v jaké situaci?	Nesekala.	Nesekala.	Nesekala.
Vyhovovalo vám ovládání vozidla? Co byste změnil?	Zvýšit rychlost couvání.	V pořádku.	V pořádku.
Vyhovovaly vám zbraně? Co byste změnil?	Přidat nový druh zbraně (raketa, kulomet, ..)	Víc typů zbraní.	Víc typů zbraní.
Jak hodnotíte herní mapu a co byste případně změnil na jejím rozložení?	Přidat víc míst s obilím.	Rozmístit balíky obilí na více míst na herní mapě.	V pořádku.
Byl průběh hry dostatečně plynulý? Narušovalo něco váš požitek ze hry?	Kamera se příliš "houpe", když se vozidlo pohybuje.	Nic nenarušovalo herní zážitek.	Nic nenarušovalo herní zážitek.
Existuje nějaká herní mechanika, kterou byste rádi ve hře viděli? Co můžeme zlepšit?	Přidat šít (časově nebo jinak omezený).	Víc typů zbraní.	Statistiky na konci hry.